CAB FARE PREDICTION

SAI KARHTIK

OCTOBER 2019

Contents

1. **Introduction**
   - **1.1** Problem Statement .....
   - **1.2** Data .....

2. **Methodology** .....
   - **2.1 Pre Processing** .....
     - **2.1.1 Variable identification** .....
     - **2.1.2 Data Cleaning** .....
     - **2.1.3 Feature Engineering** .....
     - **2.1.4 Missing Value Analysis and Treatment** .....
     - **2.1.5 Visualization** .....
       - A) Uni variate Analysis .....
       - B) Bi variate Analysis .....
     - **2.1.6 Outlier Analysis and Treatment** .....
     - **2.1.7 Feature Selection**
       - A) Correlation
       - B) Feature Importance .....
     - **2.1.8 Feature Scaling** .....
   - **2.2 Modelling** .....
     - **2.2.1 Model Selection**
       - A) LINEAR REGRESSION
       - B) DECISION TREES
       - C) RANDOM FORESTS
       - D) RIDGE REGRESSION .....
     - **2.2.2 Visualizing models**
       - A) Prediction Plots .....

3. **Conclusion**
   - **3.1 Model Evaluation** .....
     - **3.1.1** Mean Absolute Error(MAE) .....
     - **3.1.2** Mean Squared Error(MSE) .....
     - **3.1.3** Mean Squared Error(RMSE) .....
     - **3.1.4** R2 .....
     - **3.1.5** Model Score .....
   - **3.2 Model Selection** .....
   - **3.3 Execution** .....

Chapter 1

1. Introduction

Aim: To predict cab fare amount based on Given: Pick-up date and time, pick-up and drop coordinates, number of passengers

Dataset: A dataset of 16000 observations of a pilot program conducted in in New York area is given.

Problem Statement: Predict the fare_amount of renting a cab, given pickup & dropoff coordinates & number of passengers Model to be developed: Because we are required to predict a continuous value, we will be building a regression model.

Data*:* As the dataset given has dependent and independent values, it will come under supervise Machine learning. Our task is to build Regression models which will help us predicting the fare for our cab which depends on the factors provided. Given below is a sample of the data set that we are using for our prediction.

This dataset contains 07 variables in which 6 are independent variables and 1 (Fare_amount) is dependent variable.

Variable	**Explanation**
fare_amount	Float amount of the ride.
pickup_datetime	timestamp value indicating when the cab ride started.
pickup_longitude	float for longitude coordinate of where the cab ride started.
pickup_latitude	float for latitude coordinate of where the cab ride started
dropoff_longitude	float for longitude coordinate of where the cab ride ended.
dropoff_latitude	float for latitude coordinate of where the cab ride ended.
passenger_count	an integer indicating the number of passengers in the cab ride.

In [230]:
```
cabfare_train.head(10)
```

Out[230]:

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0
1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2.0
3	7.7	2012-04-21 04:30:42 UTC	-73.987130	40.733143	-73.991567	40.758092	1.0
4	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1.0
5	12.1	2011-01-06 09:50:45 UTC	-74.000964	40.731630	-73.972892	40.758233	1.0

Chapter 2

2. Methodology

2.1 Pre Processing: Before we proceeding to create our model on top of the provided data. It is necessary to do Exploratory Data Analysis. EDA is very first and necessary step to take before proceeding further. As the result depends on the data, EDA makes sure the quality of input data is high which will lead to high quality results. We can perform EDA as follows:

2.2.1 Variable Identification: In Order to understand the data, we need to first, Identifying Predictor (Input) and Target (output) variables. Then, Identifying the data type and category of the variables

Types of Variable: Our Target Variable is 'fare_amount', and Predictor variables are (pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, passenger_count).

DataTypes: Categorical(passenger_count), Numeric(fare_amount), factor(pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude), datetime(pickup_datetime).

```
fare_amount                      float64
pickup_datetime      datetime64[ns, UTC]
pickup_longitude                 float64
pickup_latitude                  float64
dropoff_longitude                float64
dropoff_latitude                 float64
passenger_count                 category
dtype: object
```

We have converted the data as per our requirement

NOTE: I made passenger_count to category after making it to Int and cleaning it.

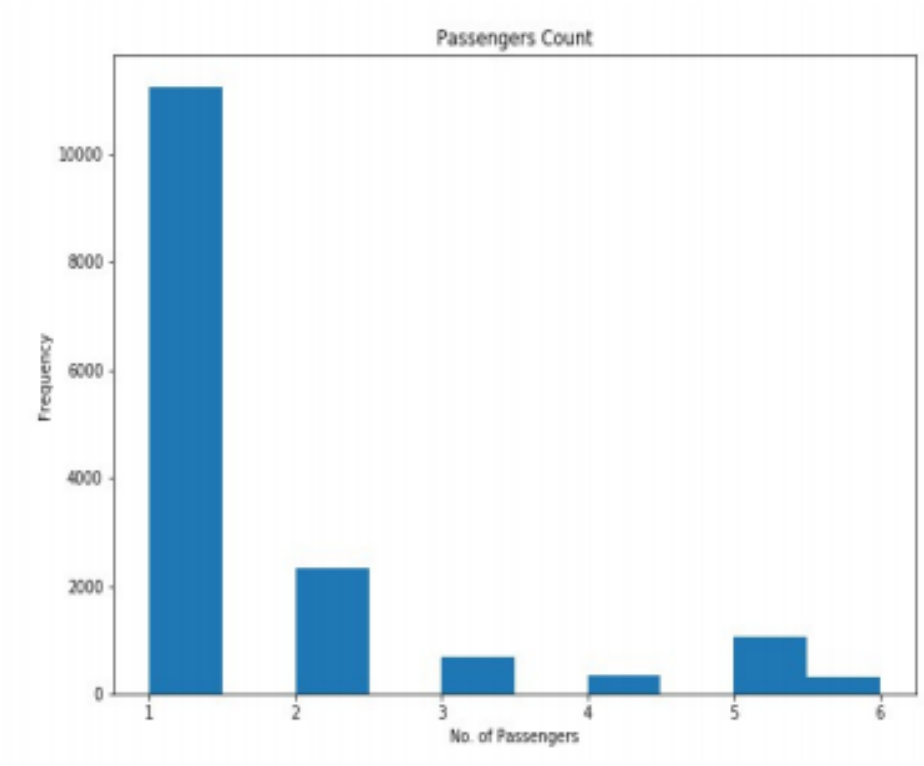
Variable Categories: Categorical (passenger_count), Continuous (pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude)

Before I made passanger_count to category, our data looked like below :

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	16043	16067	16067.000000	16067.000000	16067.000000	16067.000000	16012.000000
unique	468	16021	NaN	NaN	NaN	NaN	NaN
top	6.5	2013-06-06 19:47:00 UTC	NaN	NaN	NaN	NaN	NaN
freq	759	2	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	-72.462787	39.914725	-72.462328	39.897906	2.625070
std	NaN	NaN	10.578384	6.826587	10.575062	6.187087	60.844122
min	NaN	NaN	-74.438233	-74.006893	-74.429332	-74.006377	0.000000
25%	NaN	NaN	-73.992156	40.734927	-73.991182	40.734651	1.000000
50%	NaN	NaN	-73.981698	40.752603	-73.980172	40.753567	1.000000
75%	NaN	NaN	-73.966838	40.767381	-73.963643	40.768013	2.000000
max	NaN	NaN	40.766125	401.083332	40.802437	41.366138	5345.000000

After I made passanger_count to category our data looks like

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	16042.000000	16066	16067.000000	16067.000000	16067.000000	16067.000000	16012.0
unique	NaN	16020	NaN	NaN	NaN	NaN	27.0
top	NaN	2010-03-10 14:09:00+00:00	NaN	NaN	NaN	NaN	1.0
freq	NaN	2	NaN	NaN	NaN	NaN	11259.0
first	NaN	2009-01-01 01:31:49+00:00	NaN	NaN	NaN	NaN	NaN
last	NaN	2015-06-30 22:42:39+00:00	NaN	NaN	NaN	NaN	NaN
mean	15.015004	NaN	-72.462787	39.914725	-72.462328	39.897906	NaN
std	430.460945	NaN	10.578384	6.826587	10.575062	6.187087	NaN
min	-3.000000	NaN	-74.438233	-74.006893	-74.429332	-74.006377	NaN
25%	6.000000	NaN	-73.992156	40.734927	-73.991182	40.734651	NaN
50%	8.500000	NaN	-73.981698	40.752603	-73.980172	40.753567	NaN
75%	12.500000	NaN	-73.966838	40.767381	-73.963643	40.768013	NaN
max	54343.000000	NaN	40.766125	401.083332	40.802437	41.366138	NaN



2.2.2 Data Cleaning: We can clearly observe from Summary in R and Describe Function in Python that

- Passenger counts of maximum values is very high . Passenger count is too high as we know cab can accommodate max 8 passenger if consider its SUV.
- Pickup\drop off longitude and latitude is not under 90 and 180 which is not possible as per geographical information. Distance is also extremely high as per regular cab which roams within the city.
- So, to proceed further we are, Keeping fare_amount under 100 (as during visualization I realized the distribution of data is under 60 and after that just tail is stretched), Pickup\drop off longitude and latitude under under 90 and 180, passenger_count under 6. I have not dropped observation for not but just imputed it with NA. After cleaning our data looks like

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
count	16035.000000	15751.000000	15751.000000	15752.000000	15754.000000
mean	11.362008	-73.911857	40.690039	-73.906695	40.688008
std	10.789184	2.651448	2.605833	2.703000	2.624623
min	0.010000	-74.438233	-74.006893	-74.429332	-74.006377
25%	6.000000	-73.992387	40.736548	-73.991377	40.736257
50%	8.500000	-73.982040	40.753298	-73.980568	40.754220
75%	12.500000	-73.968081	40.767799	-73.965365	40.768309
max	453.000000	40.766125	41.366138	40.802437	41.366138

To remove the noisy data, manually lower & upper ranges for each feature was added into a dataframe.

- 'fare_amount' : [1, 100],
- 'pickup_longitude' : [-74.8, -72.8],
- 'pickup_latitude' : [39.45, 41.45],
- 'dropoff_longitude' : [-74.8, -72.8],
- 'dropoff_latitude' : [39.45, 41.45],
- 'passenger_count' : [1, 6], This includes some new features added over the course of time while programming.

2.2.3 Feature Engineering: Before we proceed ,

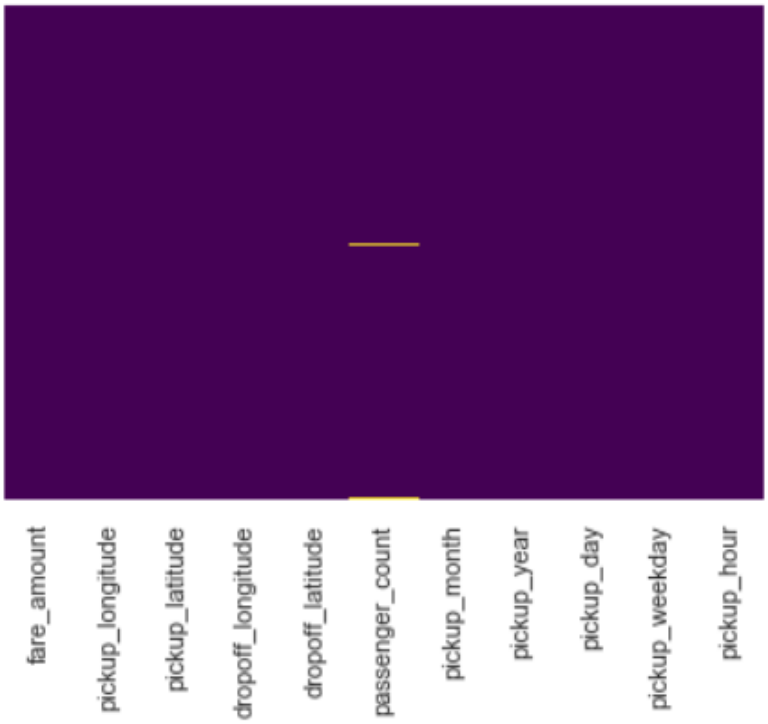
1) I have split pick_updatetime into hours, day, month and year and dropped the main variable pickup_datetime. This will help us understanding our data more efficiently.
2) I have calculated distance based on our Pickup\drop off longitude and latitude using great_circle_distance function.

After Feature engineering our data looks like this

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	great_circle_distance	year	month	weekday	hour
0	4.5	-73.844311	40.721319	-73.841610	40.712278	1.0	1.030764	2009	6	0	17
1	16.9	-74.016048	40.711303	-73.979268	40.782004	1.0	8.450134	2010	1	1	16
2	5.7	-73.982738	40.761270	-73.991242	40.750562	2.0	1.389525	2011	8	3	0
3	7.7	-73.987130	40.733143	-73.991567	40.758092	1.0	2.799270	2012	4	5	4
4	5.3	-73.968095	40.768008	-73.956655	40.783762	1.0	1.999157	2010	3	1	7
5	12.1	-74.000964	40.731630	-73.972892	40.758233	1.0	3.787239	2011	1	3	9
6	7.5	-73.980002	40.751662	-73.973802	40.764842	1.0	1.555807	2012	11	1	20
7	16.5	-73.951300	40.774138	-73.990095	40.751048	1.0	4.155444	2012	1	2	17
8	NaN	-74.006462	40.726713	-73.993078	40.731628	1.0	1.253232	2012	12	0	13

2.2.4 Missing values Analysis : Missing values occur when no data value is stored for the variable in an observation. Missing values are a common occurrence, and you need to have a strategy for treating them. A missing value can signify a number of different things in your data. Perhaps the data was not available or not applicable or the event did not happen. It could be that the person who entered the data did not know the right value, or missed filling in. Typically, ignore the missing values, or exclude any records containing missing values, or replace missing values with the mean, or infer missing values from existing values. We check for missing values in our data and came to know we have missing data in almost every variable

Before Missing Value Analysis :



	Variables	Missing_percentage
0	fare_amount	0.155598
1	pickup_longitude	0.000000
2	pickup_latitude	0.000000
3	dropoff_longitude	0.000000
4	dropoff_latitude	0.000000
5	passenger_count	0.342317
6	pickup_month	0.006224
7	pickup_year	0.006224
8	pickup_day	0.006224
9	pickup_weekday	0.006224

After Missing Value Analysis :

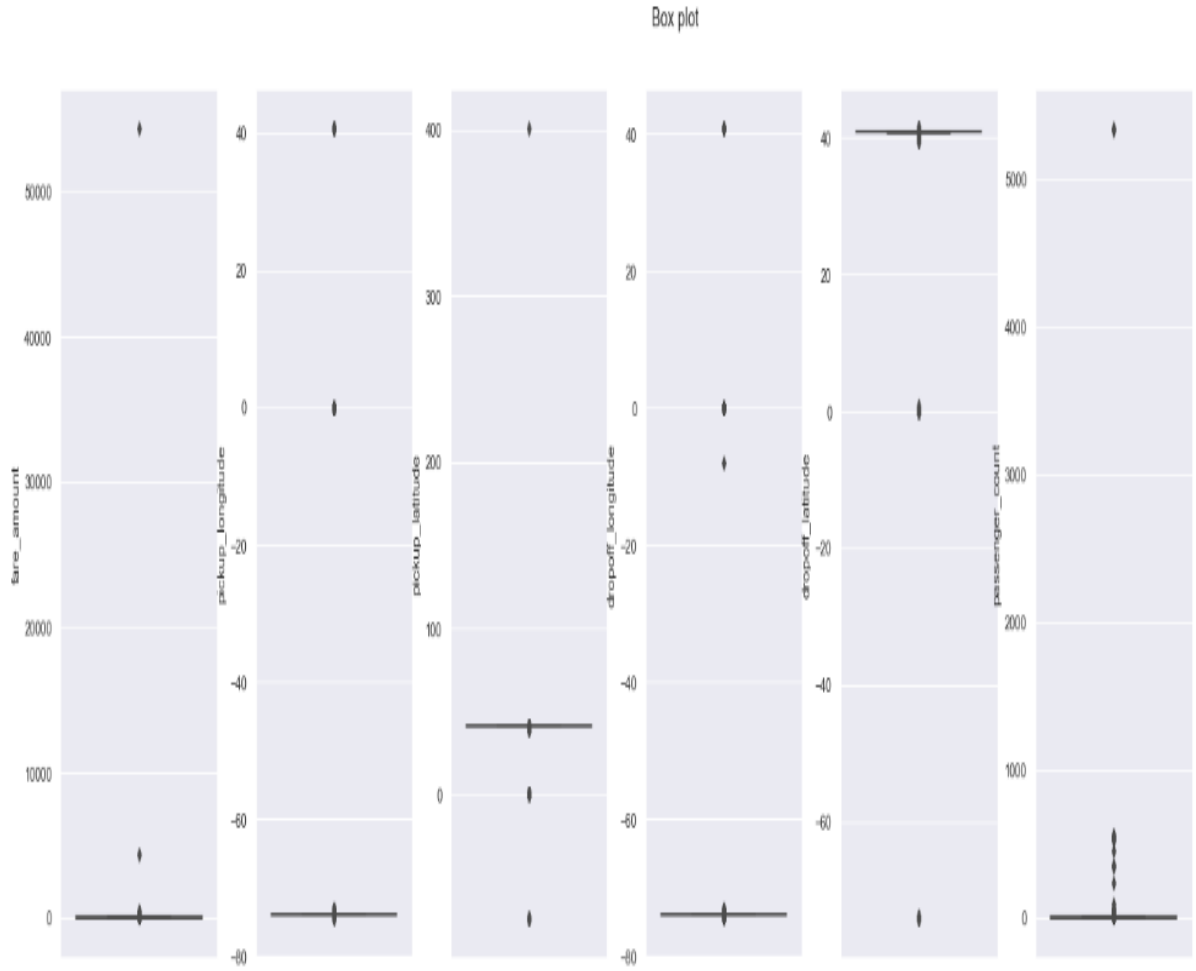
```
Number of Rows removed = 3067
```

	Variables	Missing_percentage
0	fare_amount	0.0
1	pickup_longitude	0.0
2	pickup_latitude	0.0
3	dropoff_longitude	0.0
4	dropoff_latitude	0.0
5	passenger_count	0.0
6	pickup_month	0.0
7	pickup_year	0.0
8	pickup_day	0.0
9	pickup_weekday	0.0

I have deleted observations with Missing Values in R and Python coding since distribution of missing values are same across the different variable and as a another try imputed using mean, median , KNN and other fitting formulas in Python.

2.2.5 Visualization: Exploring Variables one by one to understand central tendency, spread of the variable, distribution of each category, association and disassociation between variables at a predefined significance level.

As we see below our target variable data is under 0 to 60 and further that tail is stretched.

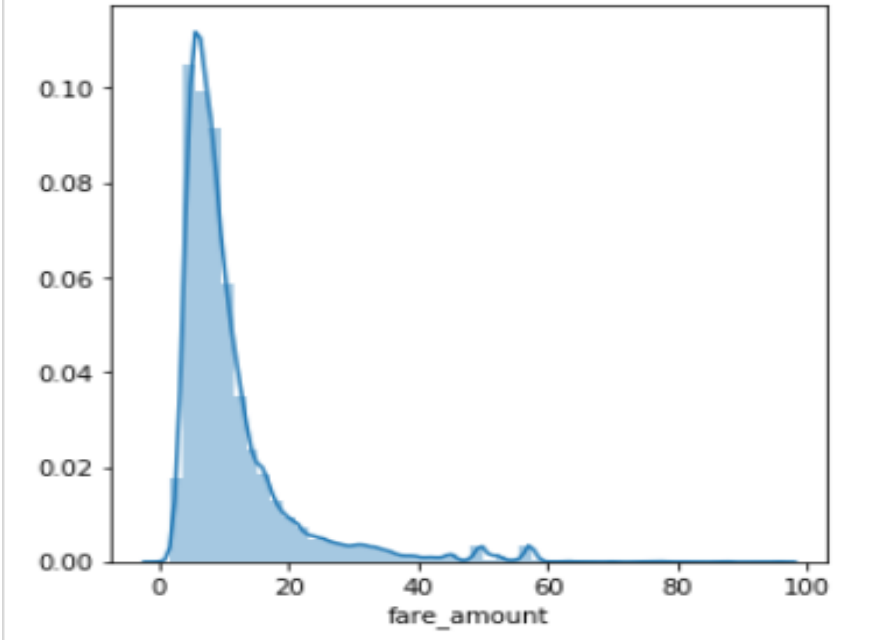


2.2.5 (A) Univariate Analysis: Checking the distribution of individual variables

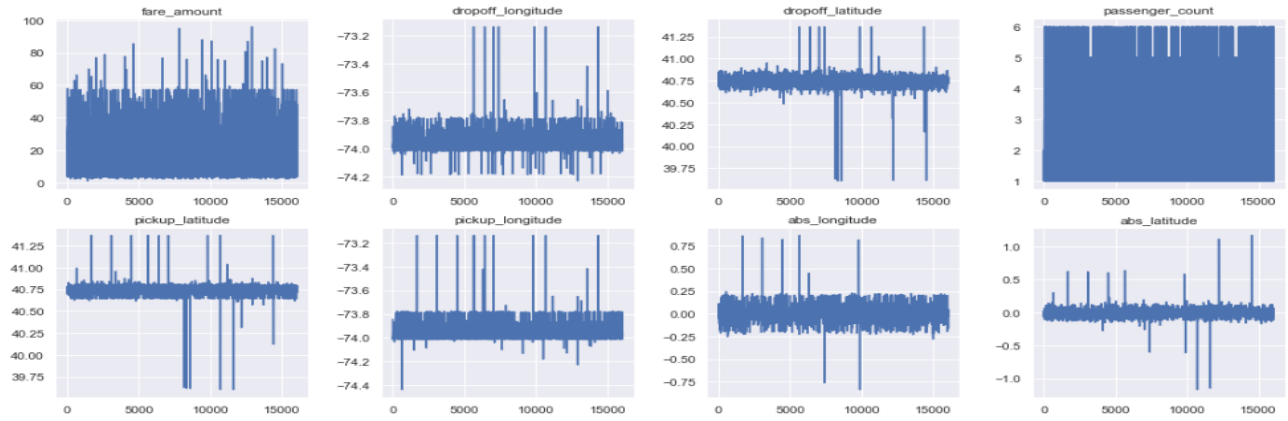
As We see below our target variable data is under 0 to 60 and further that tail is stretched

```
sns.distplot(train['fare_amount'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x
```



As per the below figure , we can understand our city is under 20 to 40 latitude, and passenger frequency of 1 is higher.



2.2.5 (B) Bi-variate Analysis: We are checking relation of variables with each other to understand the relationship of variables with our target variable

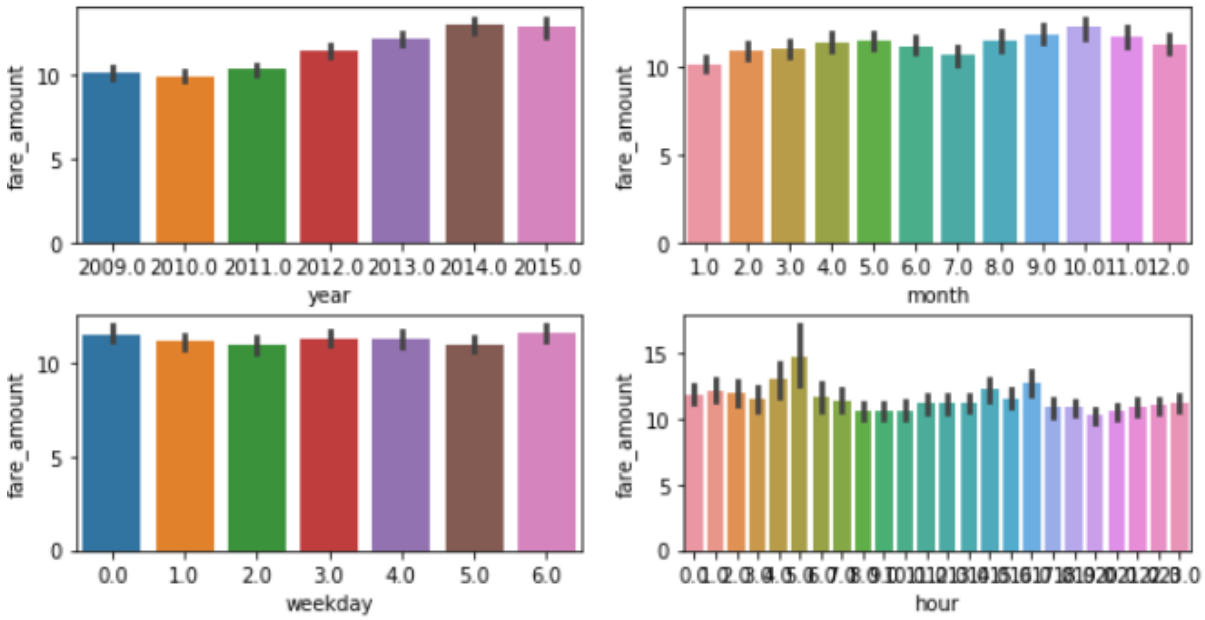
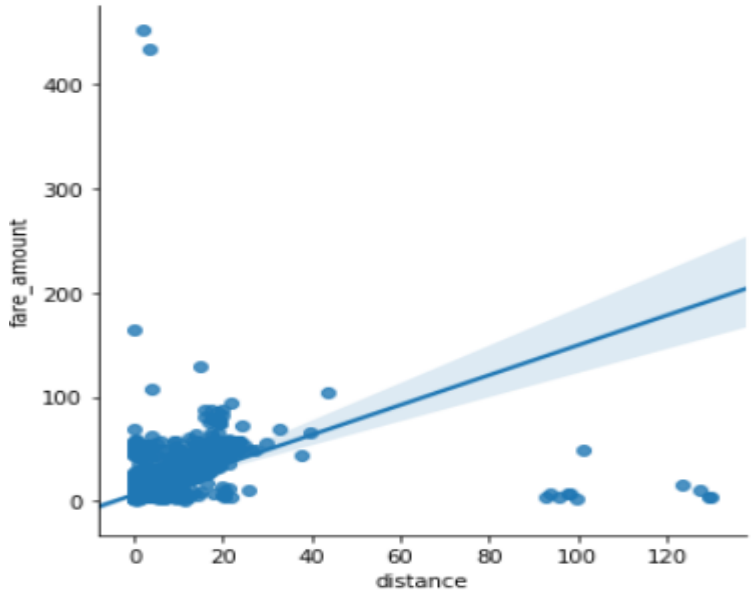
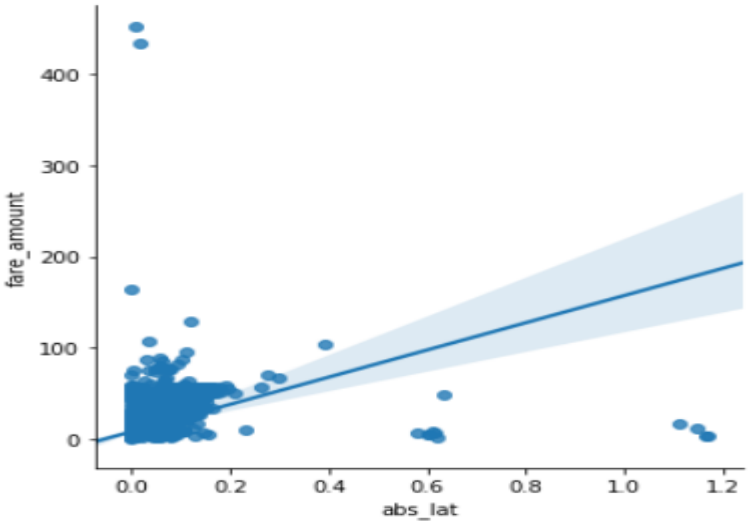


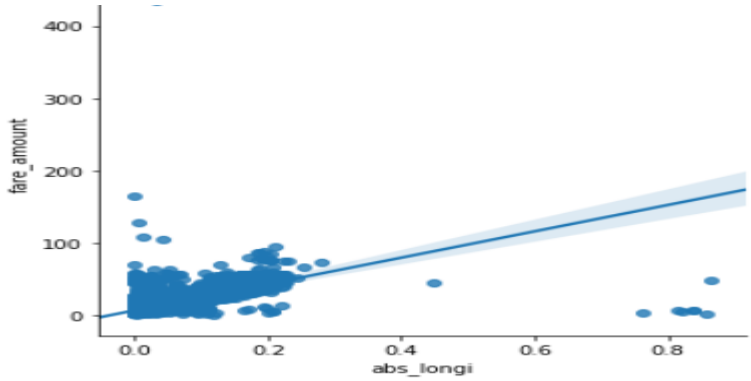
Fig 1.1 – bar pot for distribution of year, month, weekday and hour against fare_amount



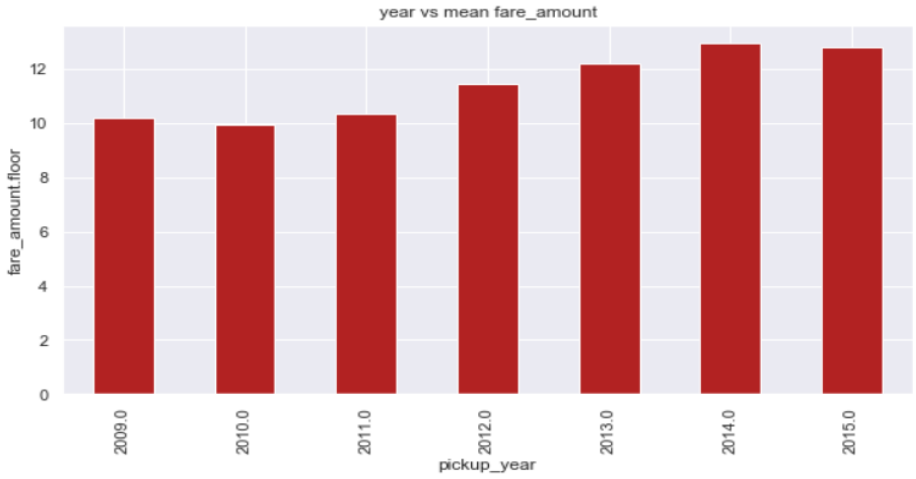
Distance v/s fareamount



Absolute latitude v/s fare amount

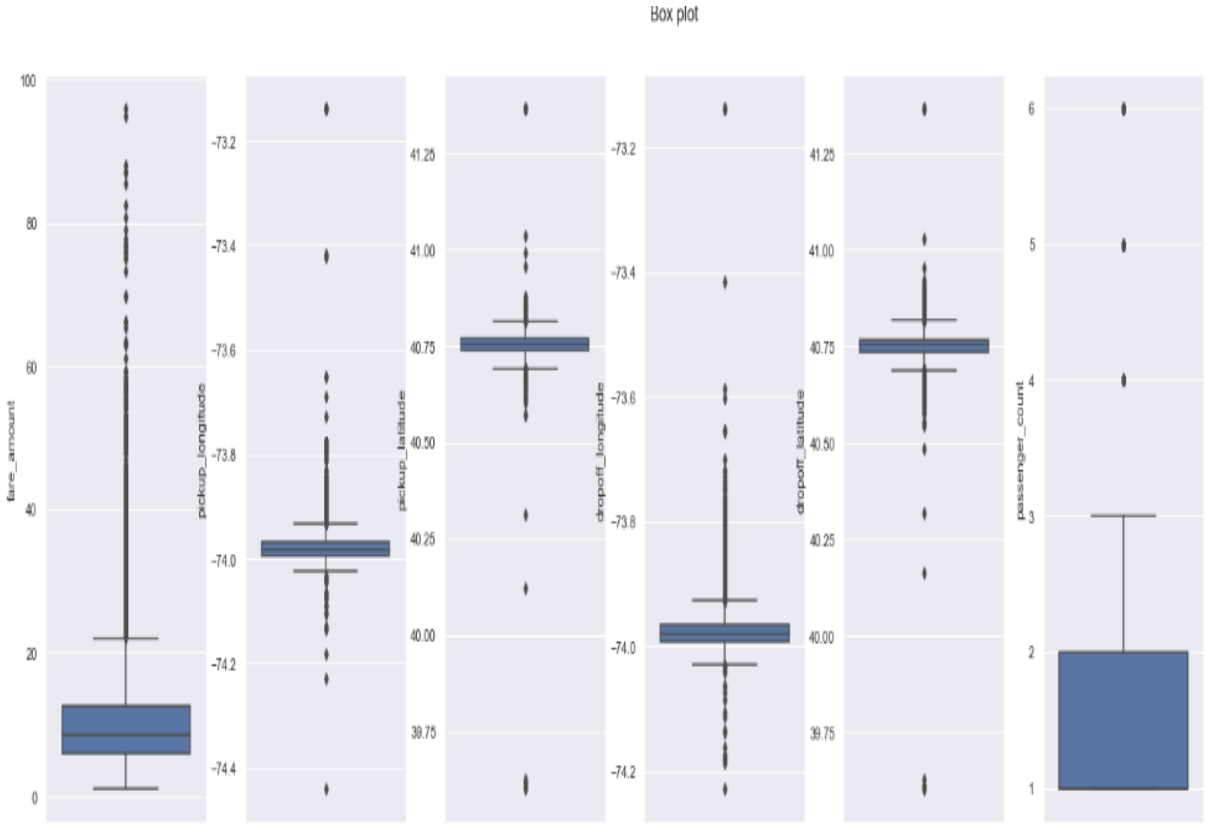


Absolute Longitude v/s fare amount plot

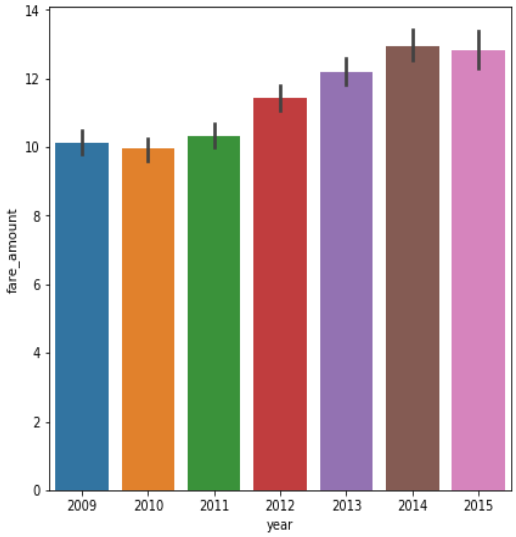


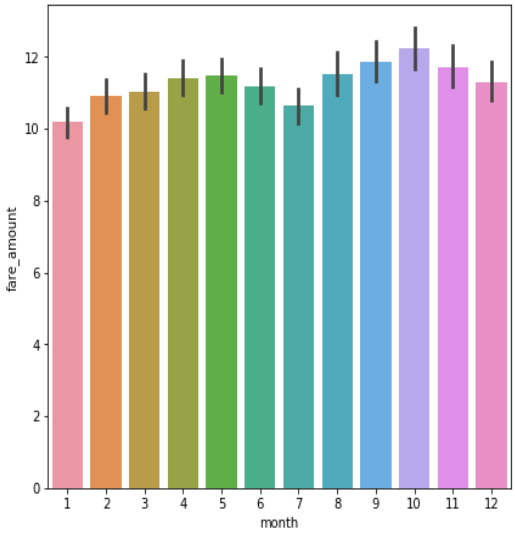
2.2.6 Outlier treatment: An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Outliers can drastically change the results of the data analysis and statistical modelling. There are numerous unfavourable impacts of outliers in the data set. It increases the error variance and reduces the power of statistical tests. If the outliers are non-randomly distributed, they can decrease normality. They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.

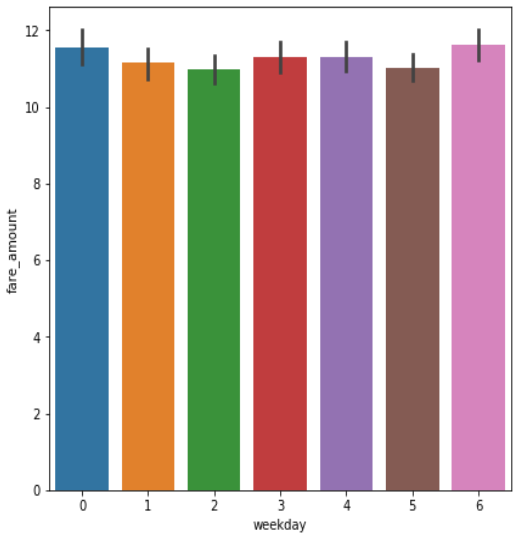
In our data, we can clearly observe from Summary in R and Describe Function in Python that Passenger counts of maximum values is very high and Pickup\drop off longitude and latitude is not under 90 and 180. Hence I have marked passenger count more than 8 to 2(to avoid outliers) as a cab can only accommodate 8 people at the max and kept Pickup\drop off longitude under 90 and 180 as per geographical information. I have also minimized Distance to 500km (but my model was highly skewed so further minimize to 100).

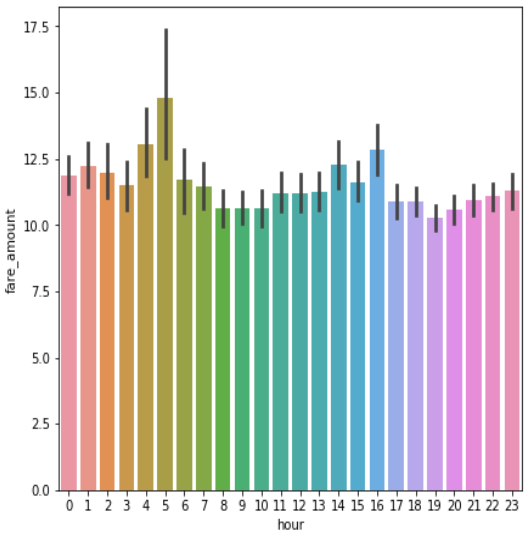


2.2.7 Feature Selection: We have converted Pickup \ drop off latitude and longitude as absolute location points and from these variables we have extracted the total distance travelled. From Pick date and Time extracted Year, Month, day, Hours. Here is some graphical representation

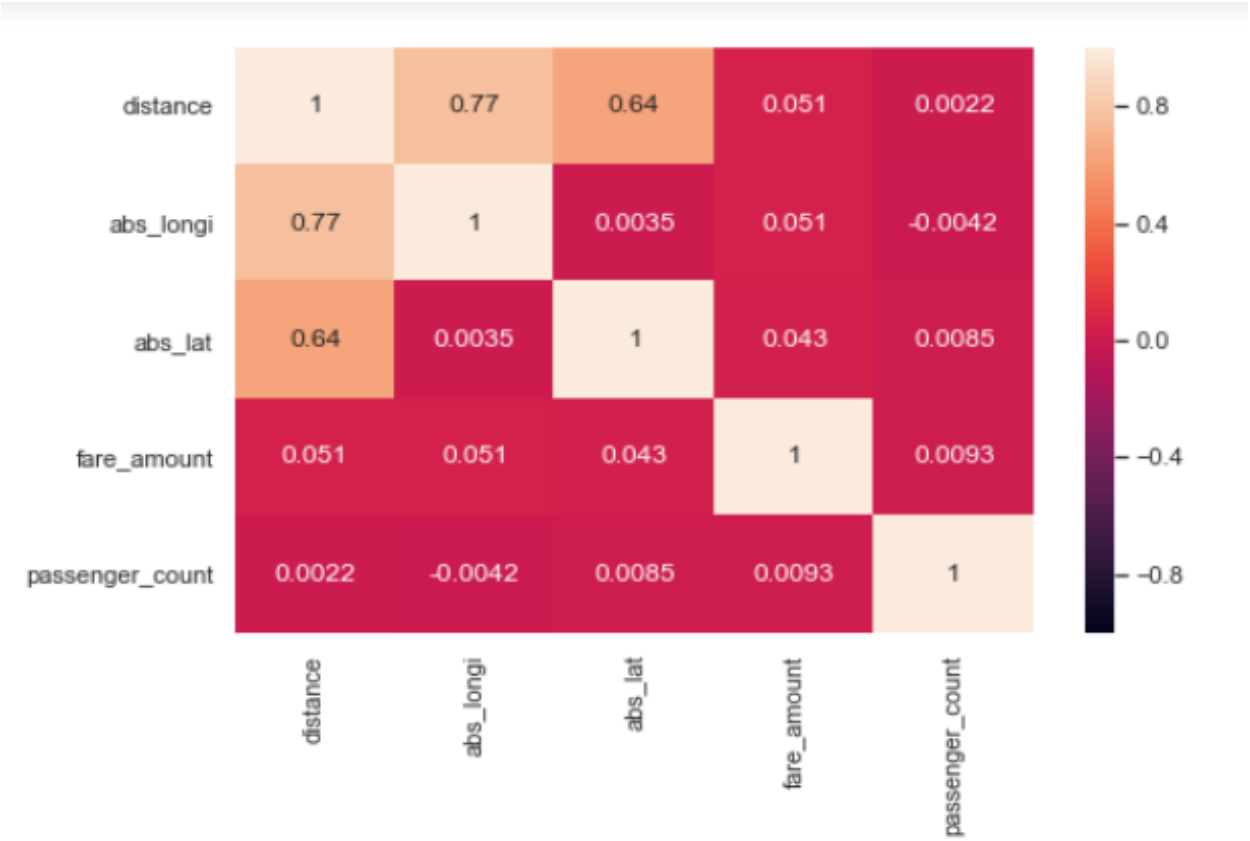






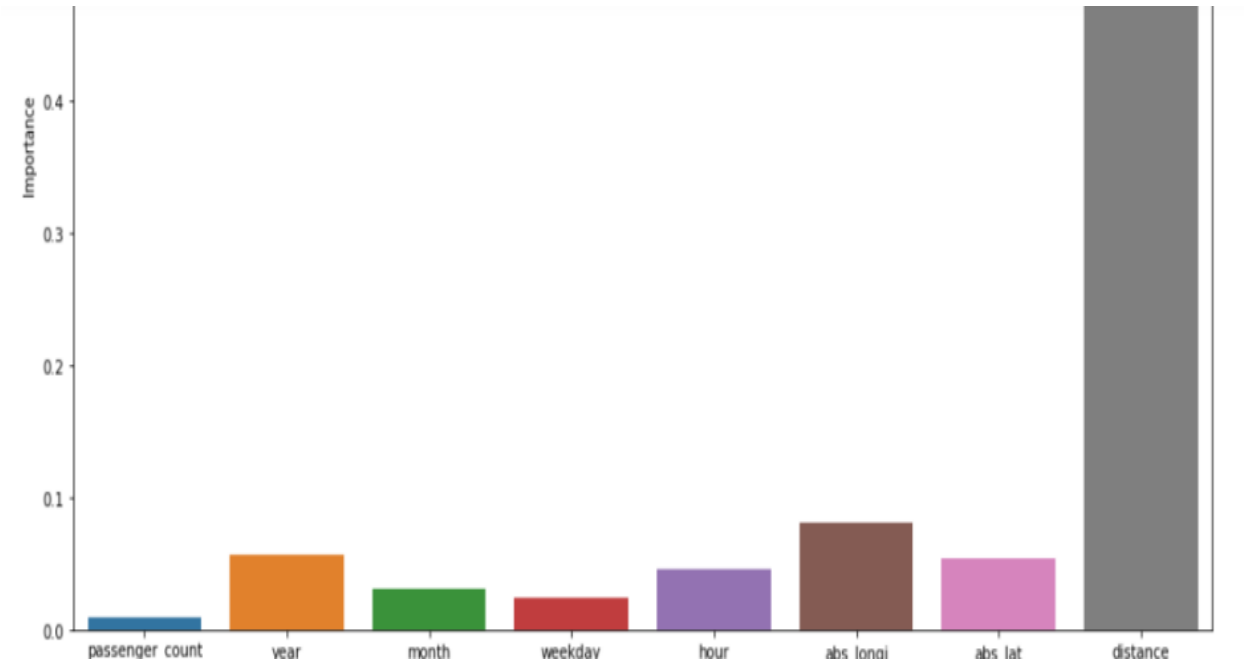


Correlation Analysis : We make heat map to understand the co relation of contiguous variable. A heat map is a graphical representation of data where the individual values contained in a matrix are represented as *colours*. Here each numerical variable's correlation is mapped with each other's in a matrix which has been plotted in the following *heat map.*



Feature Importance: The concept is really straightforward: We measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature. A feature is "important" if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction. A feature is "unimportant" if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction.

Checking via Tree:



Here , we can see the the importance of distance is extremely high. So, instead of deleting all other variables , I am going to create out model with two inputs one with distance only and one with all the variable including distance.

2.8 Feature Scaling: Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization. Normalization also called Min-Max scaling. It is the process of reducing unwanted variation either within or between variables. Normalization brings all of the variables into proportion with one another. It transforms data into a range between 0 and 1. All our continuous variables are already normalized except the target and the distance which we took out from logi/lati variable which we prefer not to scale because its variation is spread quite widely and after scaling, the difference between the number is diminishing.

Checking for Skewness and Kurtosis: Skewness is usually described as a measure of a dataset's symmetry – or lack of symmetry. A perfectly symmetrical data set will have a skewness of 0. If the skewness is between -0.5 and 0.5, the data are fairly symmetrical If the skewness is between -1 and – 0.5 or between 0.5 and 1, the data are moderately skewed. If the skewness is less than -1 or greater than 1, the data are highly skewness.

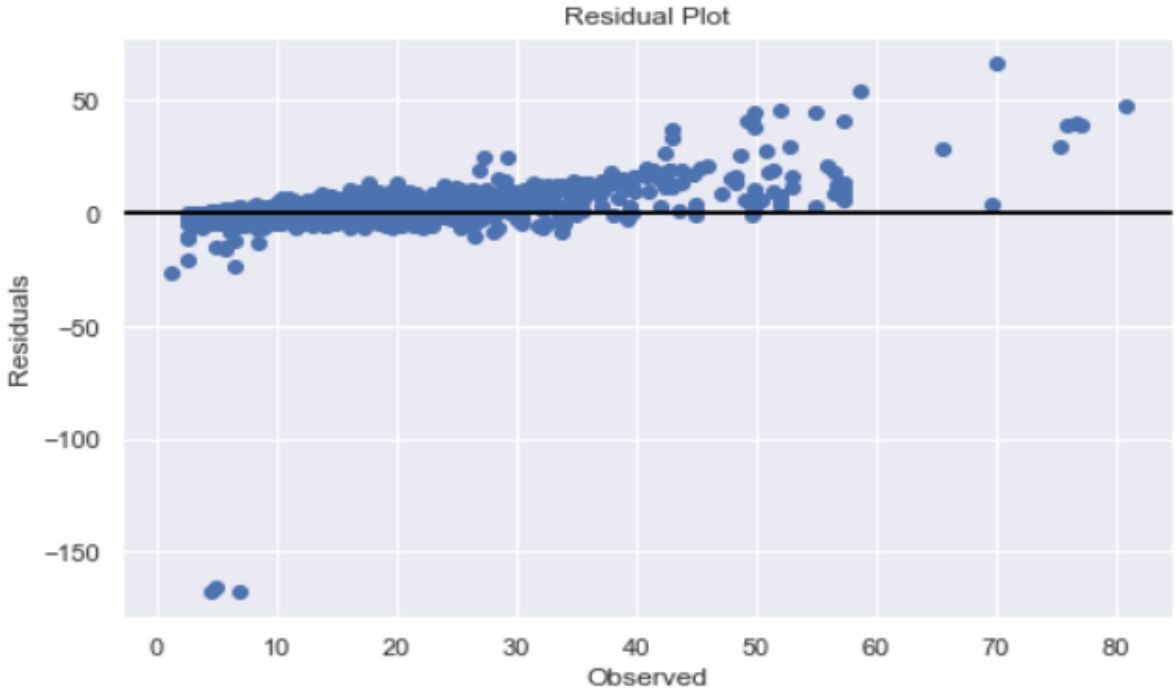
```
: print("Skewness: %f" % train['fare_amount'].skew())
  print("Kurtosis: %f" % train['fare_amount'].kurt())

  Skewness: 0.504713
  Kurtosis: -0.373433
```

Our data is little skewed and sample does not look Gaussian. Skewed data messes up the pred ictive model and it affects the regression intercept, coefficients associated with the model.
So, to reduce the Skewness I am log transforming our data. After log transform our data looks like below.

2.2 Modeling

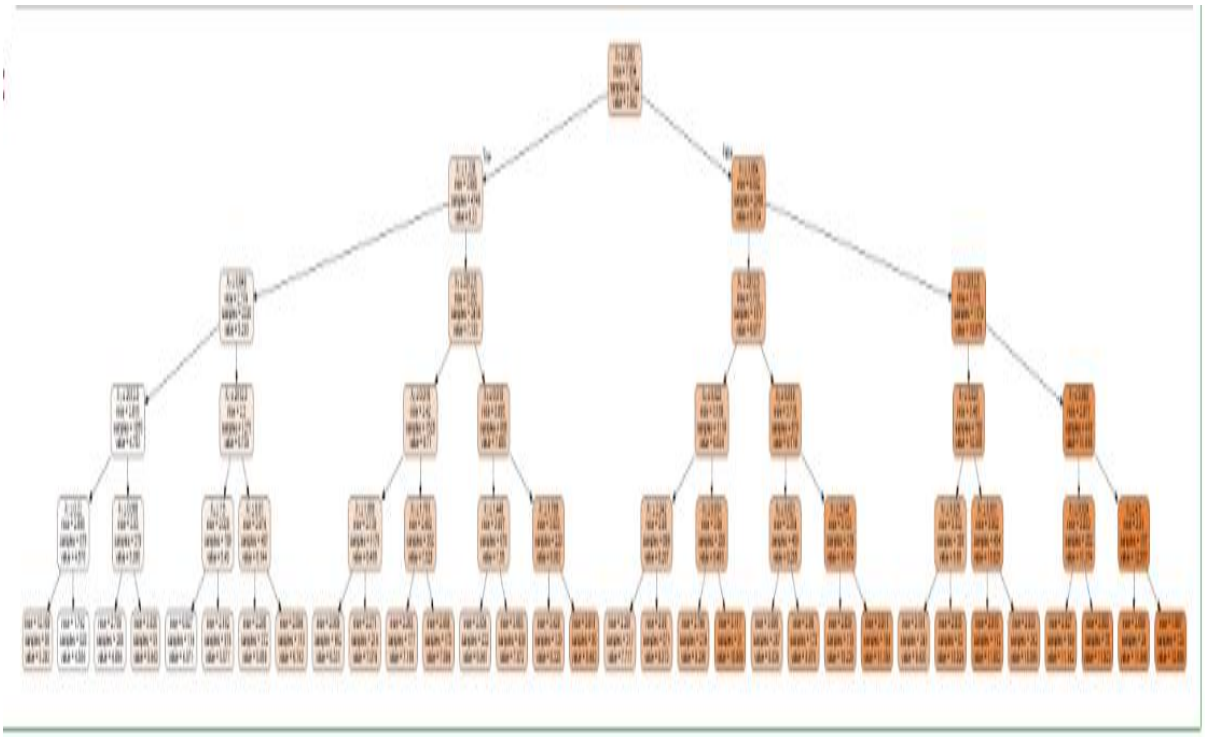
2.2.1 Model Selection: For modelling, we are going to use some famous models to our data-set and will conclude the result according to it.

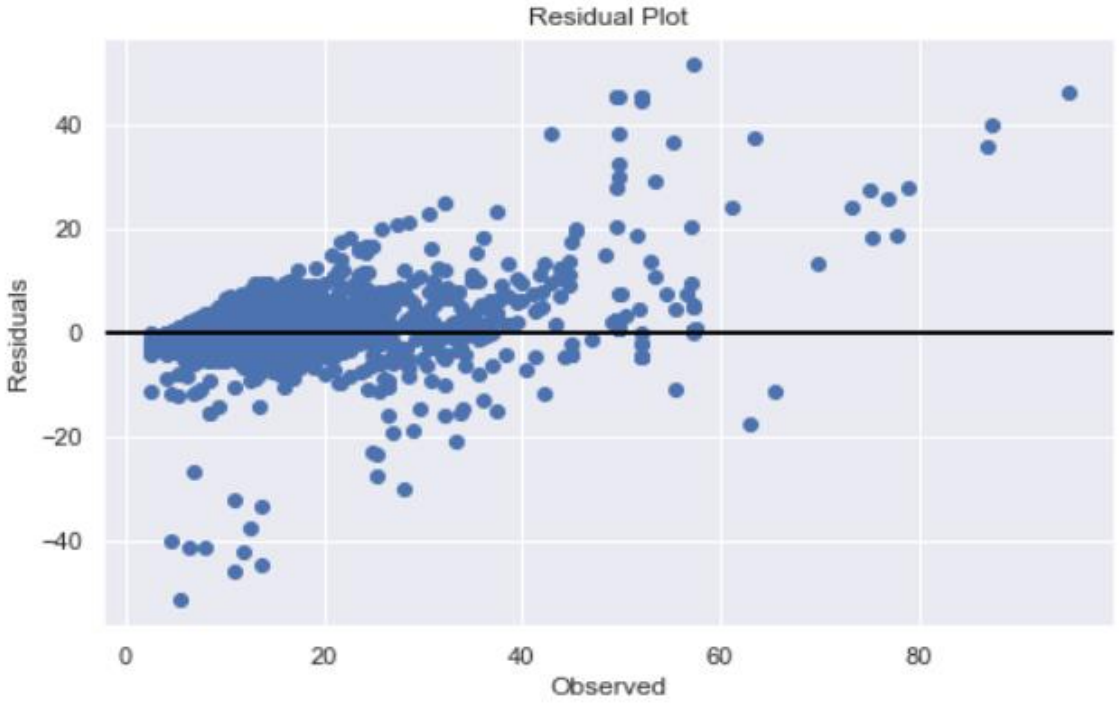
a) Linear Regression: Linear regression is the most basic type of regression and commonly used predictive analysis. Linear regression is an approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. --> Following is the summary of the Linear model:



b) Decision Tree: Decision tree is a rule. Each branch connects nodes with “and” and multiple branches are connected by “or”. It can be used for classification and regression. It is a supervised machine learning algorithm. Accept continuous and categorical variables as independent variables. Extremely easy to understand by the business users. Split of decision tree is seen in the below tree.

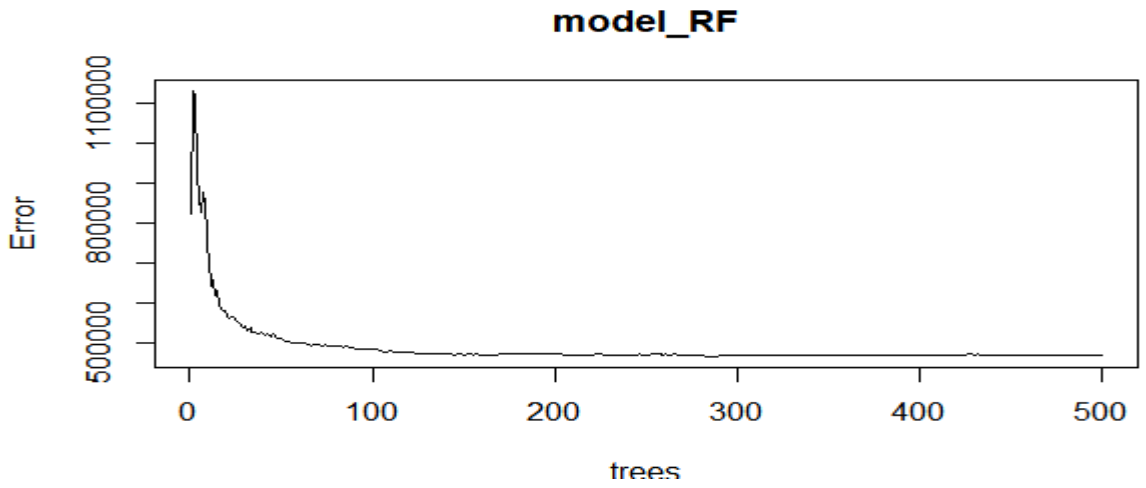
Here , is the tree for our mode

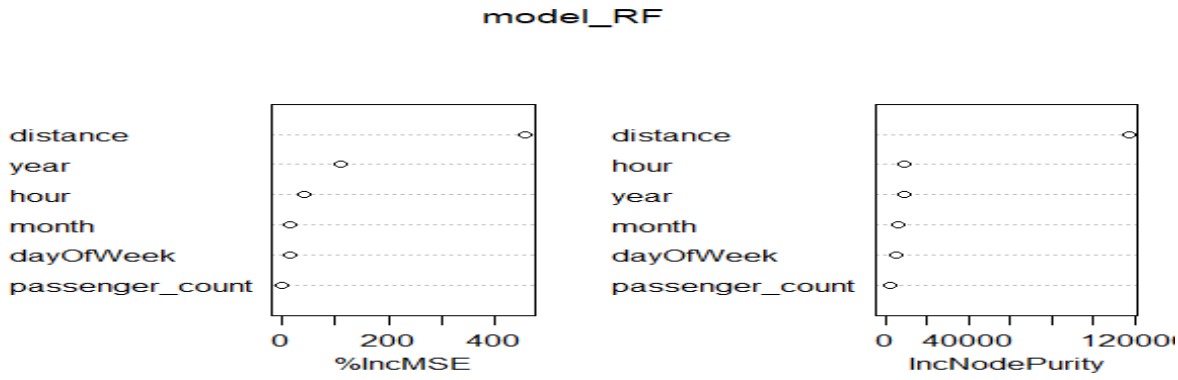



Residual Plot
Residuals
Observed
40
20
0
−20
−40
0
20
40
60
80

c) Random Forest: Random Forest or decision tree forests are an ensemble learning method for classification, regression and other tasks. It consists of an arbitrary number of simple trees, which are used to determine the final outcome. In the regression problem, their responses are averaged to obtain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases). The goal of using a large number of trees is to train enough that each feature has a chance to appear in several model--> As we increase the number of trees the error count decrease until a point (100 trees) and then becomes constant. Error vs number of trees to be used graph is as follows:

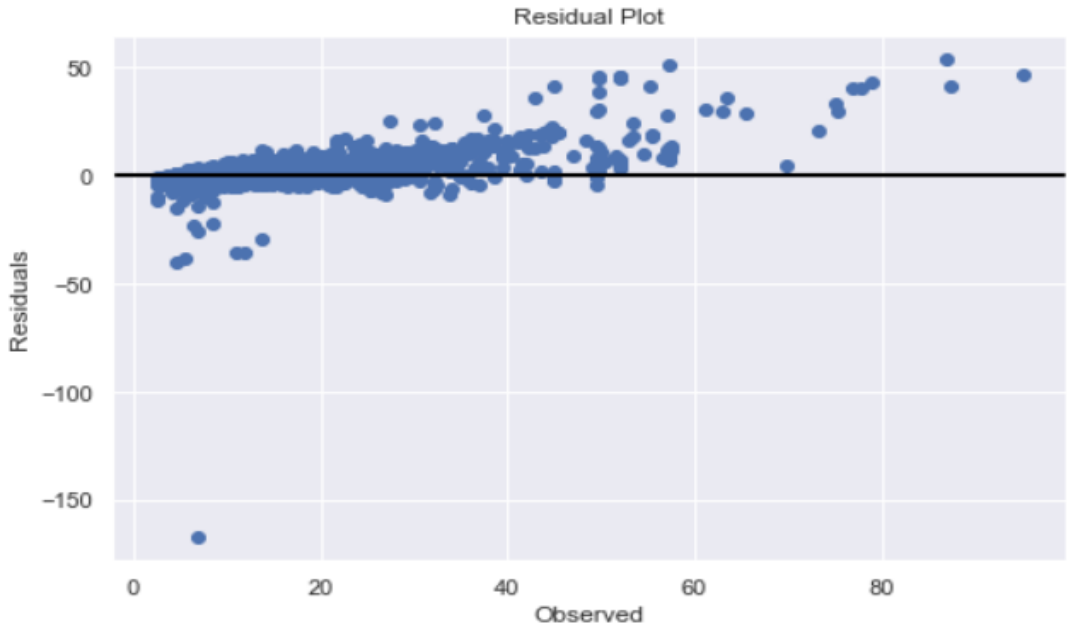
.



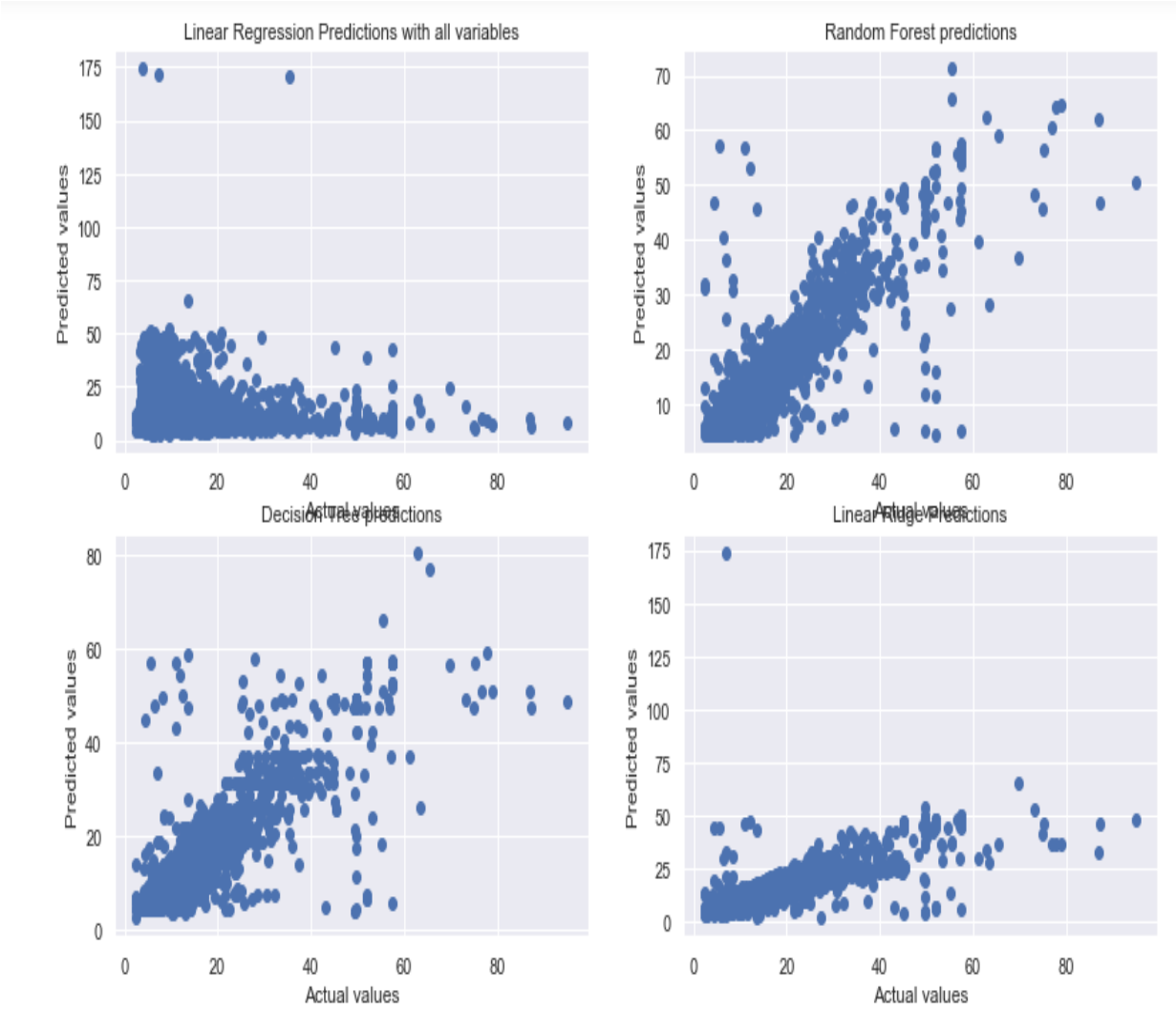


The first graph shows that if a variable is assigned values by random permutation by how much will the MSE increase. Higher the value, higher the importance. On the other hand, node purity is measured by the Gini index which is the difference between before and after split on that variable.

d) Ridge Regression: Ridge regression is the most basic type of regression and commonly used predictive analysis. Ridge regression is an approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables). The case of one explanatory variable is called Ridge regression.



2.2.2 Visualizing models : We can see the plots of our predicted model to understand it better



Chapter 3

3. Conclusion

Model Selection: We can see that all models perform comparatively on average and therefore we select random forest classifier models for better prediction. From the above plots of Actual Vs Predicted values, we can infer that values of Random forest falls on straight line indicating random forest fits better than the other three models. Also amongst the three models, Random forest has best R-sq. (Coef. of determination). Hence we'll fix Random Forest as our model.

 # LINEAR REGRESSION RESULTS

MSE: 37.11805524461228

RMSE: 6.092458883292712

MAPE: 26.982905058315783

R-Sq: 0.5734271490089067

Score: 0.67

#DECISION TREE RESULTS

MSE: 20.705571397714674

RMSE: 4.550337503714936

MAPE: 20.382260789504123

R-Sq: 0.7712133174280049

Score: 0.93

#RANDOM FOREST RESULTS

MSE: 16.801683821900976

RMSE: 4.098985706476784

MAPE: 19.364757273529392

R-Sq: 0.814349412078502

Score: 0.935

#RIDGE REGRESSION RESULTS

MSE: 29.006407119531893

RMSE: 5.385759660394427

MAPE: 27.769054587216434

R-Sq: 0.6794930441309692

Score:0.63

MSE : RANDOM FOREST<DECISION TREE<RIDGE REGRESSION<LINEAR REGRESSION

RMSE : RANDOM FOREST<DECISION TREE<RIDGE REGRESSION<LINEAR REGRESSION

MAPE : RANDOM FOREST<DECISION TREE<RIDGE REGRESSION<LINEAR REGRESSION

R-Sq: LINEAR REGRESSION<RIDGE REGRESSION<DECISION TREE<RANDOM FOREST

Score: RANDOM FOREST>DECISION TREE>LINEAR REGRESSION>RIDGE REGRESSION

By comparing the values of RMSE, MSE, MAPE, SCORE I'm choosing Random Forest Model for this dataset to predict the fare amount.

Model Selection : We can see that all models perform comparatively on average and therefore we select random forest classifier models for better prediction.

From the above plots of Actual Vs Predicted values, we can infer that values of Random forest falls on straight line indicating random forest fits better than the other three models. Also amongst the three models, Random forest has best R-sq. (Coef. of determination). Hence we'll fix Random Forest as our model.

Execution/Answer : Applying the prediction model on test data, we get below prediction distribution.